

# Vector Semantics

Assignment Project Exam Help

<https://eduassistpro.github.io/>

Add WeChat Lizhen Q edu\_assist\_pro

# Recap

---

- Language model.

$$P(x_1, x_2, \dots, x_l) = P(x_1) \prod_{i=2}^l P(x_i | x_{i-1})$$

- Kneser-Ney smoothing:  
**Assignment Project Exam Help**
- Stupid Backoff:

<https://eduassistpro.github.io/>

$$S(w_i | w_{i-k+1}^{i-1}) = \begin{cases} \frac{\text{count}(w_{i-k+1}^i)}{\text{count}(w_{i-k+1}^{i-1})} & \text{Add WeChat edu_assist}>\text{pro} \\ 0.4S(w_i | w_{i-k+2}^{i-1}) & \text{otherwise} \end{cases}$$

# Overview of the NLP Lectures

---

- Introduction to natural language processing (NLP).
- Regular expressions, sentence splitting, tokenization, part-of-speech tagging.  
[Assignment](#) [Project](#) [Exam](#) [Help](#)
- Language models <https://eduassistpro.github.io/>
- Vector semantics.  
– Multiclass logistic regression.  
[Add WeChat edu\\_assist\\_pro](#)
- Parsing.
- Compositional semantics..

# Logistic Regression for Binary Classification

---

Assignment Project Exam Help

<https://eduassistpro.github.io/>

Add WeChat edu\_assist\_pro

$$P(Y = 1|X = \mathbf{x}) = \frac{1}{1 + \exp(-(\alpha + \sum_i \beta_i x_i))}$$

# Classification of Multi-classes

---

Assignment Project Exam Help

<https://eduassistpro.github.io/>

Add WeChat edu\_assist\_pro

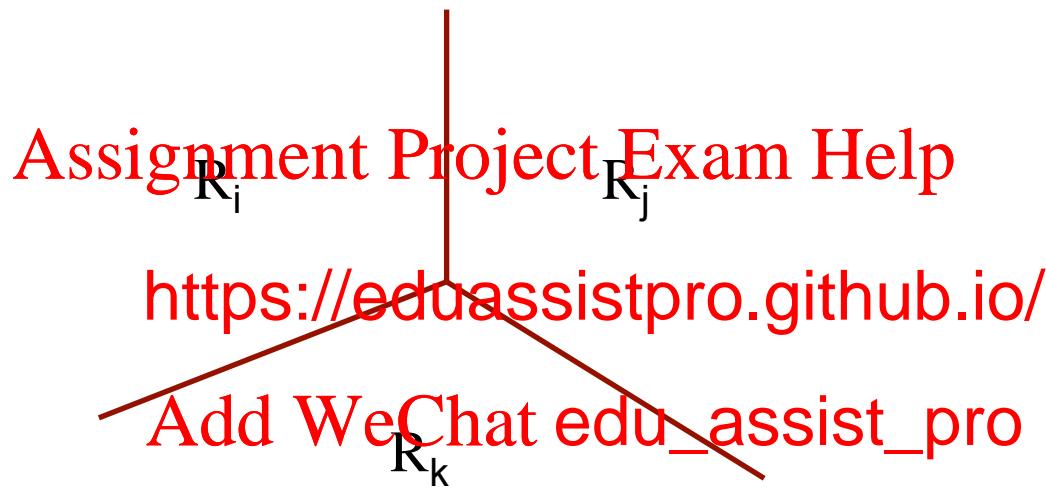
One versus rest

One versus One

# When there is only

---

- Key idea: Use K discriminant functions  $g_m(\mathbf{x})$  and pick the max.



$$g_k(\mathbf{x}) > g_i(\mathbf{x}) \text{ and } g_k(\mathbf{x}) > g_j(\mathbf{x})$$

# Softmax for Classification

---

- Definition:

$$P(Y = j | X = \mathbf{x}_i) = \frac{\exp(z_j)}{\sum_{j'=1}^K \exp(z_{j'})}$$

where  $z_m = g_m(\mathbf{x})$  (<https://eduassistpro.github.io/>)

Add WeChat `edu_assist_pro`

$g_m(\mathbf{x})$  is a linear function so that  $g_m(\mathbf{x}) = \alpha_m + \sum_k \beta_k^m x_k$ .

# Training of Multiclass LR

---

- Training data  $\{(\mathbf{x}_i, y_i)\}_{i=1}^N$ .

$$\hat{\theta} = \arg \max_{\theta} \log \left[ \prod_{j=1}^N P(y_j | \mathbf{x}_j; \theta) \right]$$

Assignment Project Exam Help

Rewrite e<https://eduassistpro.github.io/>

if  $y_1 = 3$ , then  $\mathbf{t}_1 = (0,0,1,0,0)$   
if  $y_2 = 1$ , then  $\mathbf{t}_2 = (1,0,0,0,0)$

# Training of Multiclass LR

---

- Training data  $\{(\mathbf{x}_i, y_i)\}_{i=1}^N$ .

$$\hat{\theta} = \arg \max_{\theta} \log \left[ \prod_{j=1}^N P(y_j | \mathbf{x}_j; \theta) \right]$$

<https://eduassistpro.github.io/>

$$= \arg \max_{\theta} \log \left[ \prod_{j=1}^N k | \mathbf{x}_j; \theta )^{t_{jk}} \right]$$

# Training of Multiclass LR

---

- Training data  $\{(\mathbf{x}_i, y_i)\}_{i=1}^N$ .

$$\hat{\theta} = \arg \max_{\theta} \log [P(y_j | \mathbf{x}_j; \theta)]$$

<https://eduassistpro.github.io/>

$$= \arg \max_{\theta} \log [ \sum_{j=1}^N \sum_{k=1}^K t_{jk} \log P(Y = k | \mathbf{x}_j; \theta) ]$$

$$= \arg \max_{\theta} \sum_{j=1}^N \sum_{k=1}^K t_{jk} \log P(Y = k | \mathbf{x}_j; \theta)$$

Cross Entropy

# Loss Functions (Classification)

---

Given:  $D = \{(\mathbf{x}_1, y_1), \dots, (\mathbf{x}_m, y_m)\}$

Loss:

$$L_D(\theta) = -\log P(D|\theta)$$

<https://eduassistpro.github.io/>

$$\arg \min_{\theta} L_D(\theta) = \arg \max$$

Usually minimize negative log-likelihood.

# Parameter Learning

---

Have a parametric loss function  $L(w_0, w_1, \dots, w_n)$   
where  $\theta = (w_0, w_1, \dots, w_n)$

Aim to  $\min_w L(w_0, w_1, \dots, w_n)$

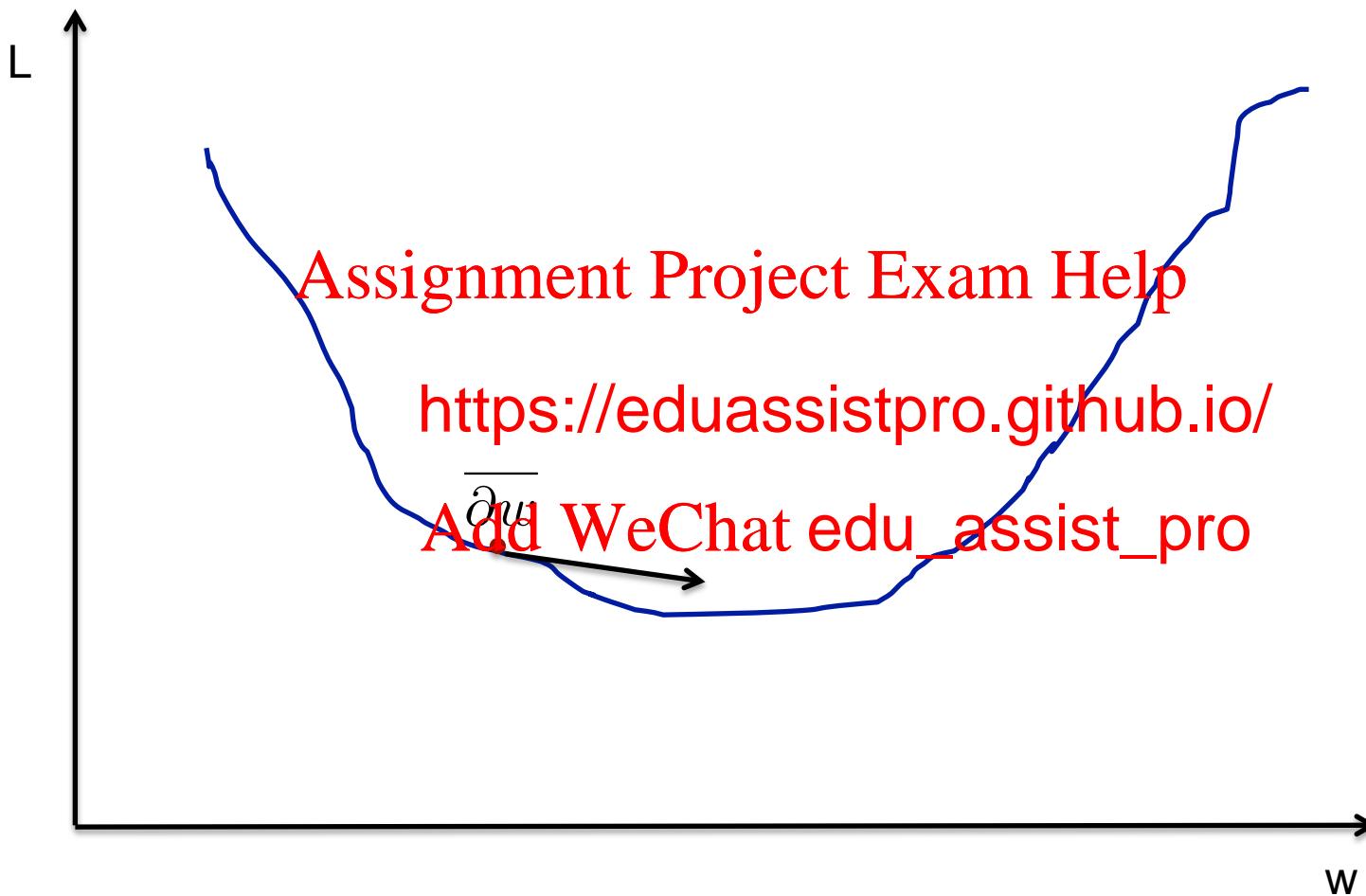
<https://eduassistpro.github.io/>

Family of Stochastic gradient Add WeChat edu\_assist\_pro  
GD):

1. Start with some  $w_0, w_1, \dots, w_n$ .
2. **Repeat:** update  $w_0, w_1, \dots, w_n$  to reduce  $L(w_0, \dots, w_n)$   
**Until:** reach a local minimum.

# Gradient Descent

---



# Descent Methods

---

Procedure:

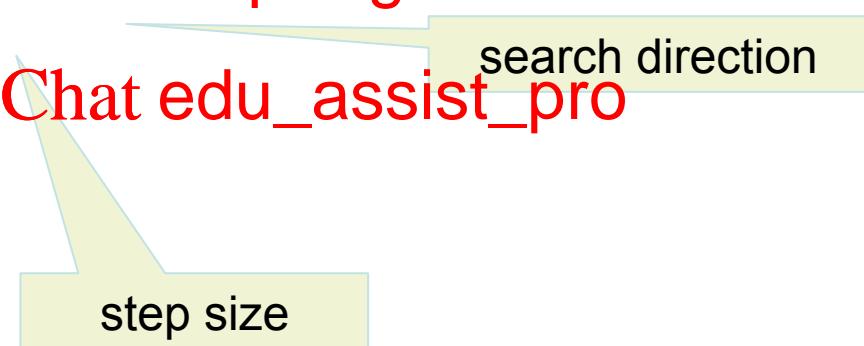
1. Start with some  $w^0$ .

- Assignment Project Exam Help
2. Repeat: up

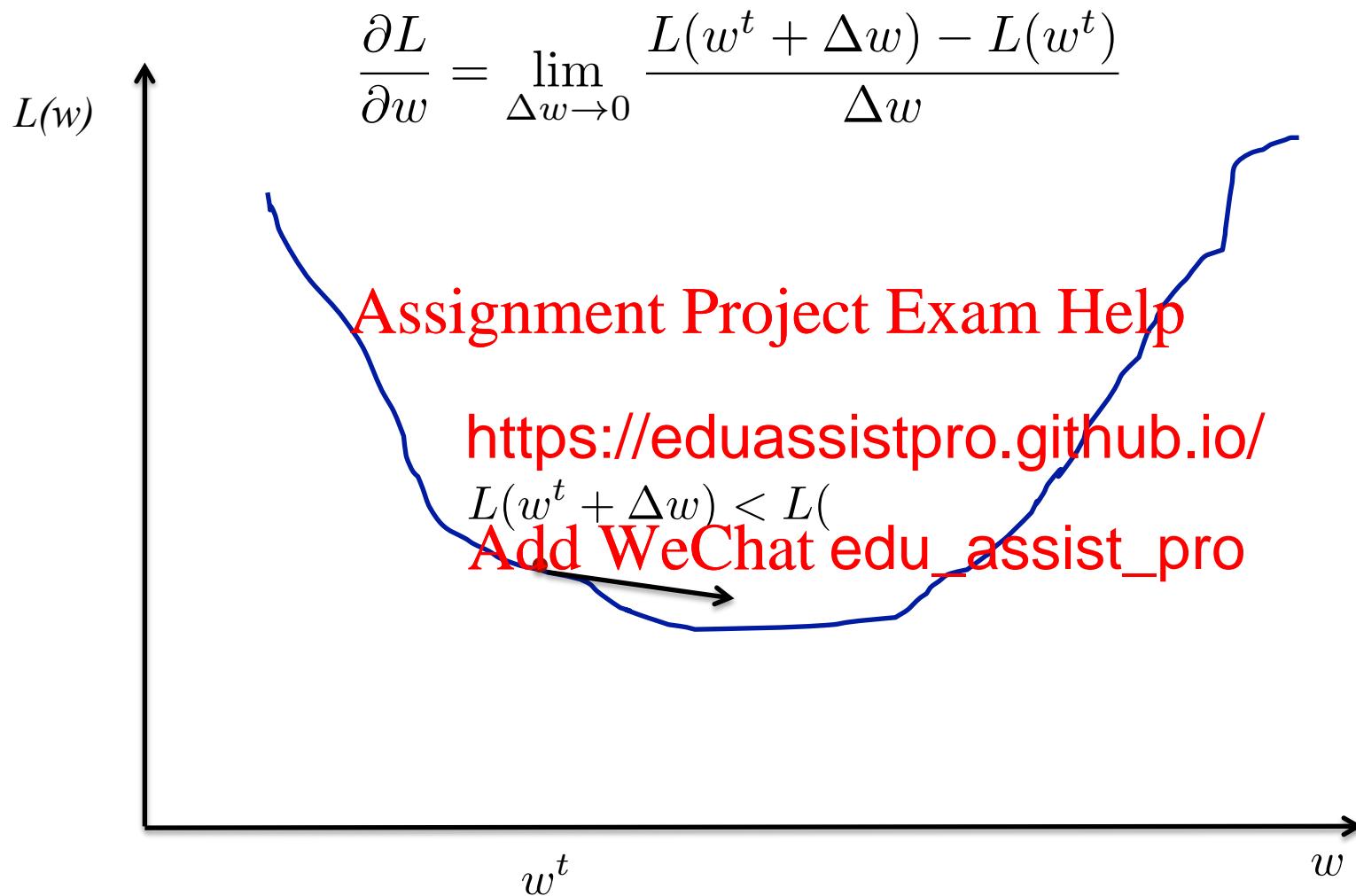
$w^{t+}$  <https://eduassistpro.github.io/>

Add WeChat edu\_assist\_pro

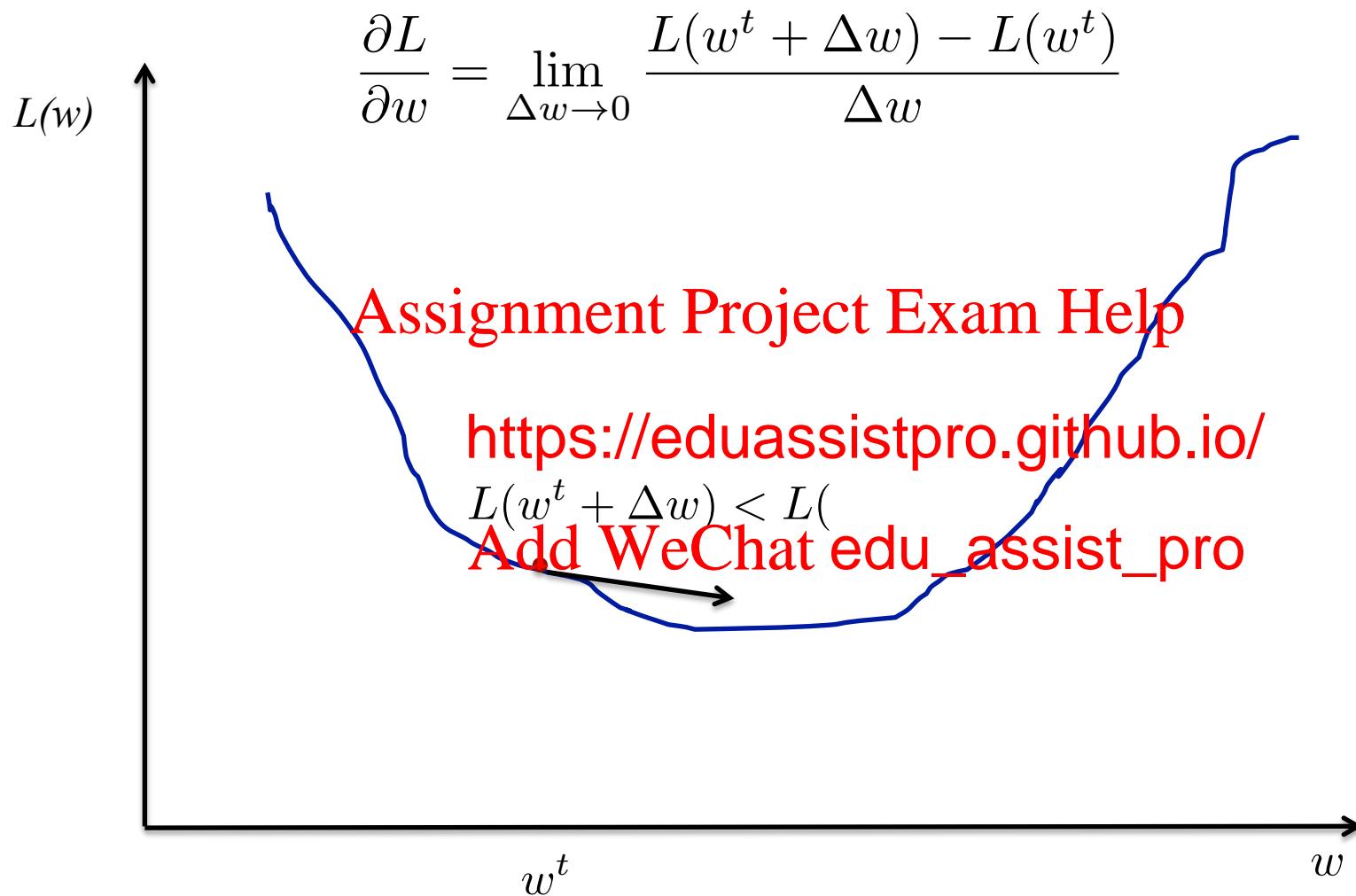
Until convergence.



# Gradient Descent (Univariate)

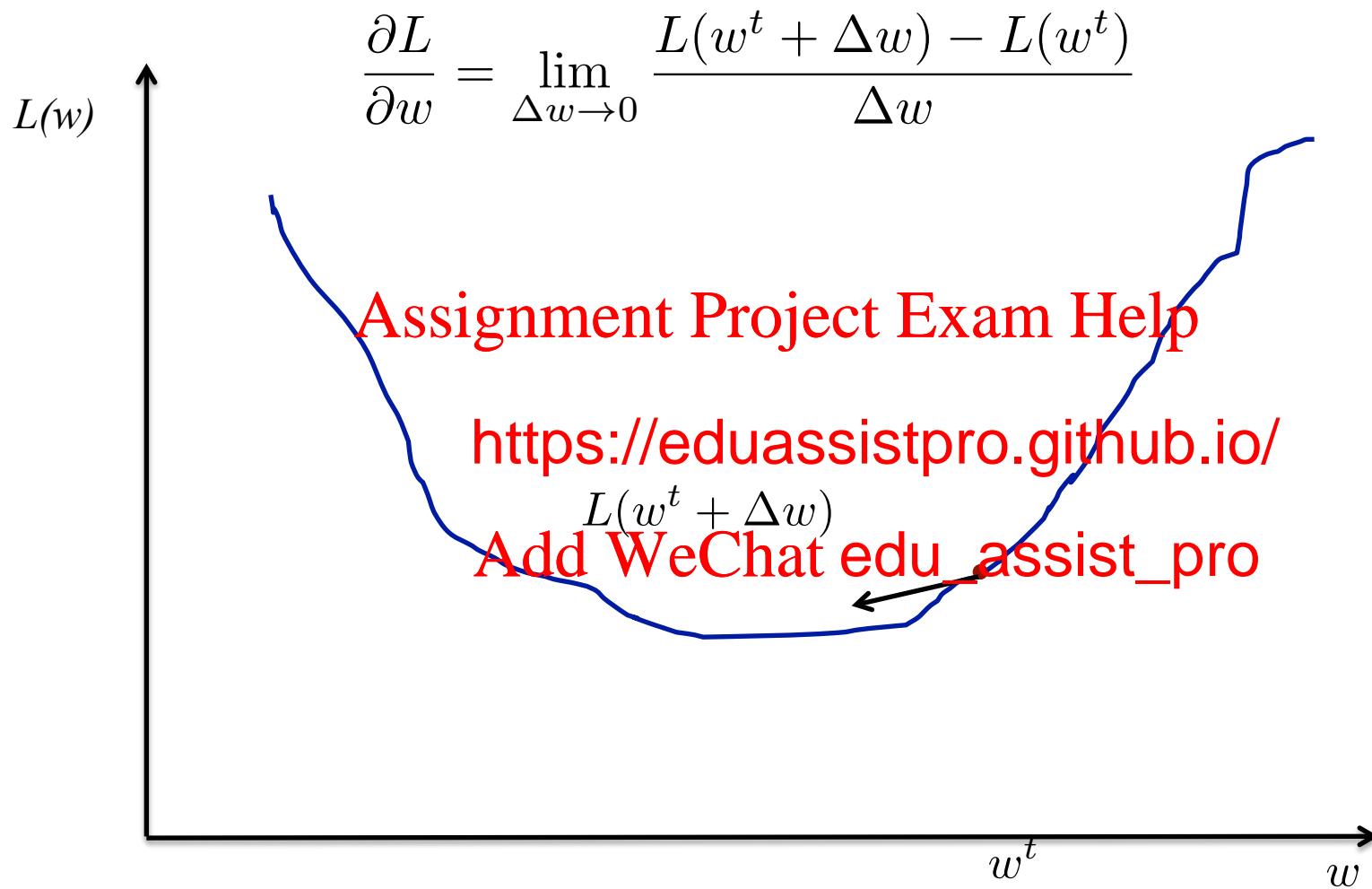


# Gradient Descent (Univariate)



$$w^t - \alpha \left( \frac{\partial L}{\partial w} \right) > w^t$$

# Gradient Descent (Univariate)



$$w^t - \alpha \left( \frac{\partial L}{\partial w} \right) < w^t$$

# Gradient Descent

---

Procedure:

1. Start with some  $w^0$ .

Assignment Project Exam Help

2. Repeat: up

<https://eduassistpro.github.io/>

Add WeChat edu\_assist\_pro

Until convergence.

# Gradient Descent

---

Procedure:

1. Start with some  $\mathbf{w}^0$ .

Assignment Project Exam Help

2. Repeat: up

<https://eduassistpro.github.io/>

Add WeChat edu\_assist\_pro  
Until convergence.

Example: 
$$\frac{L(\mathbf{w}^t)}{\mathbf{w}} = \frac{1}{N} \sum_{i=1}^N ((\mathbf{w}^t)^T \mathbf{x}_i - y_i) \mathbf{x}_i$$

# Search for Optimal Parameters

---

$$\min_{\mathbf{w}} L(\mathbf{w})$$

Assignment Project Exam Help

<https://eduassistpro.github.io/>

$L(w_0, w_1)$

Add WeChat edu\_assist\_pro

$w_0$

$w_1$

# Gradient Descent

---

$$\min_{\mathbf{w}} L(\mathbf{w})$$

Assignment  $\frac{\partial L(\mathbf{w})}{\partial \mathbf{w}}$  Project Exam Help

<https://eduassistpro.github.io/>

$L(w_0, w_1)$  Add WeChat edu\_assist\_pro

$w_0$   $w_1$

# SGD with Mini-Batch

---

- Use mini-batch:

REPEAT  $k$  epochs:

SHUFFLE training dataset  
**Assignment Project Exam Help**

FOR

<https://eduassistpro.github.io/>

$w_i^{t+1} = w_i^t - \alpha \nabla_{\theta} L(w)$

early stopping evaluated on validation set

# Overview of the NLP Lectures

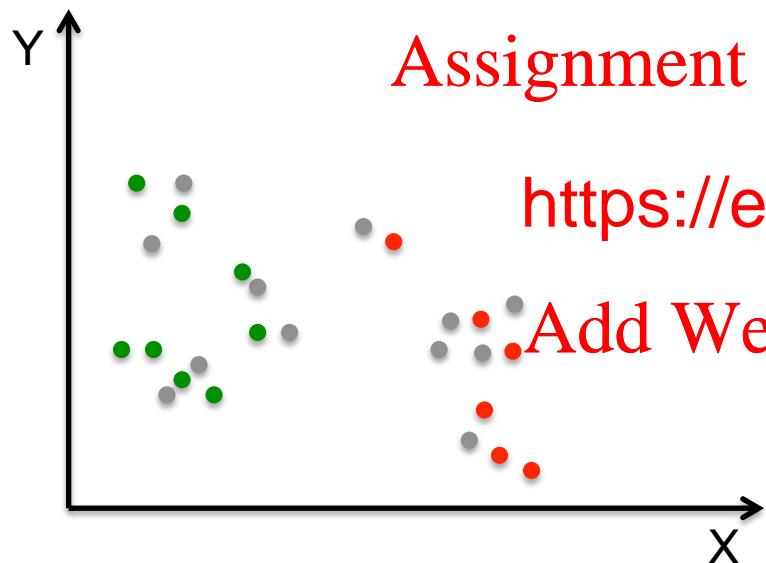
---

- Introduction to natural language processing (NLP).
- Regular expressions, sentence splitting, tokenization, part-of-speech tagging.  
**Assignment Project Exam Help**
- Language models <https://eduassistpro.github.io/>
- Vector semantics  
**Add WeChat edu\_assist\_pro**
  - Multiclass logistic regression.
  - Feedforward neural networks.
- Parsing.
- Compositional semantics.

# Nonlinearity

---

Linear Classification



Nonlinear Classification



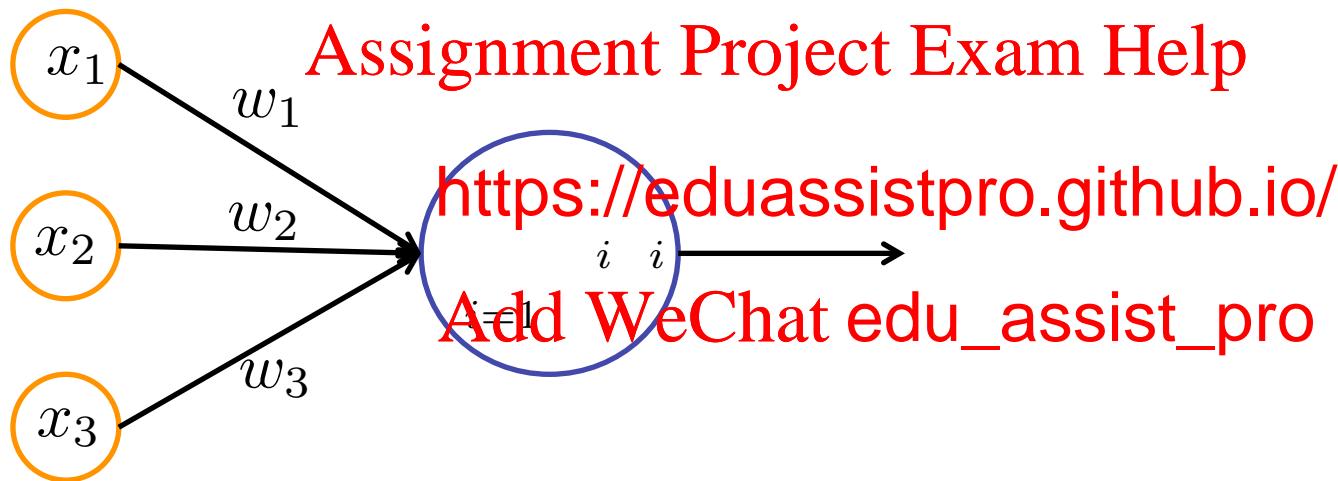
Assignment Project Exam Help

<https://eduassistpro.github.io/>

Add WeChat edu\_assist\_pro

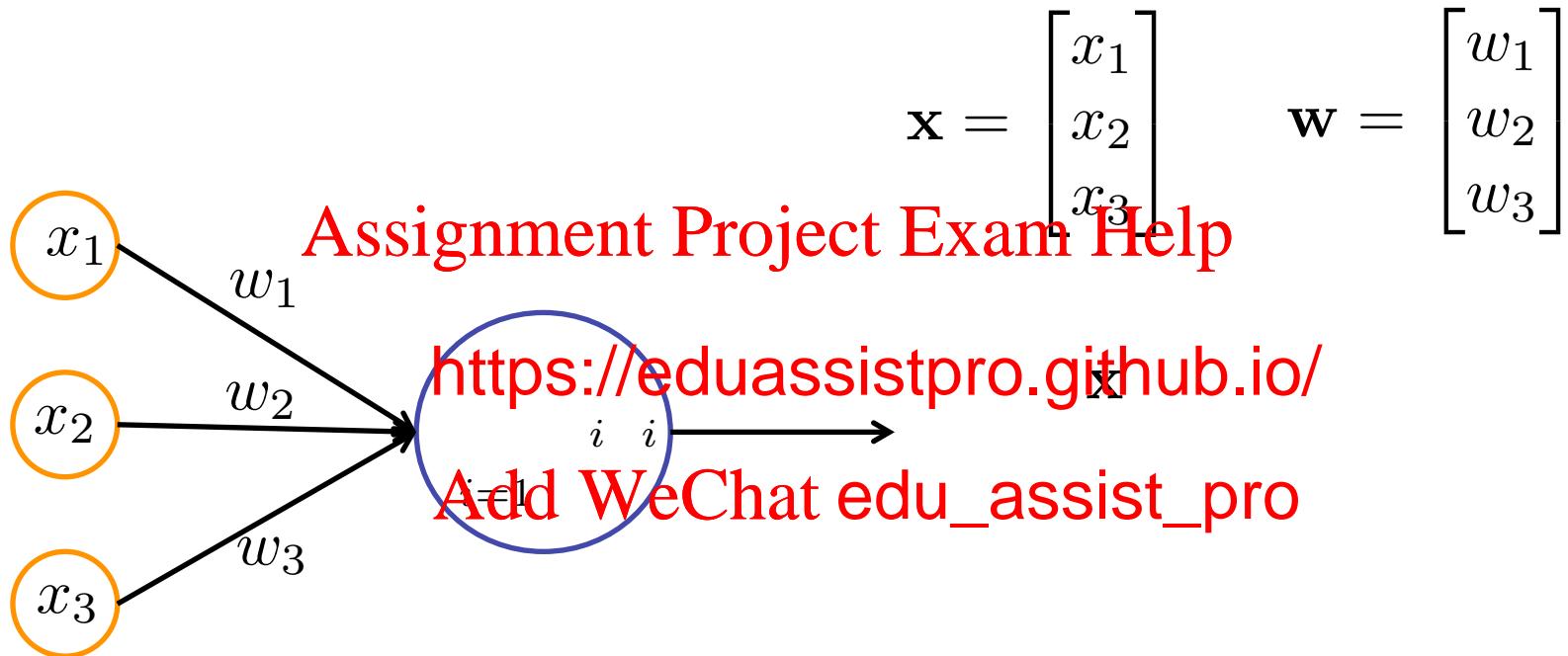
# Artificial Neuron

---



# Artificial Neuron

---

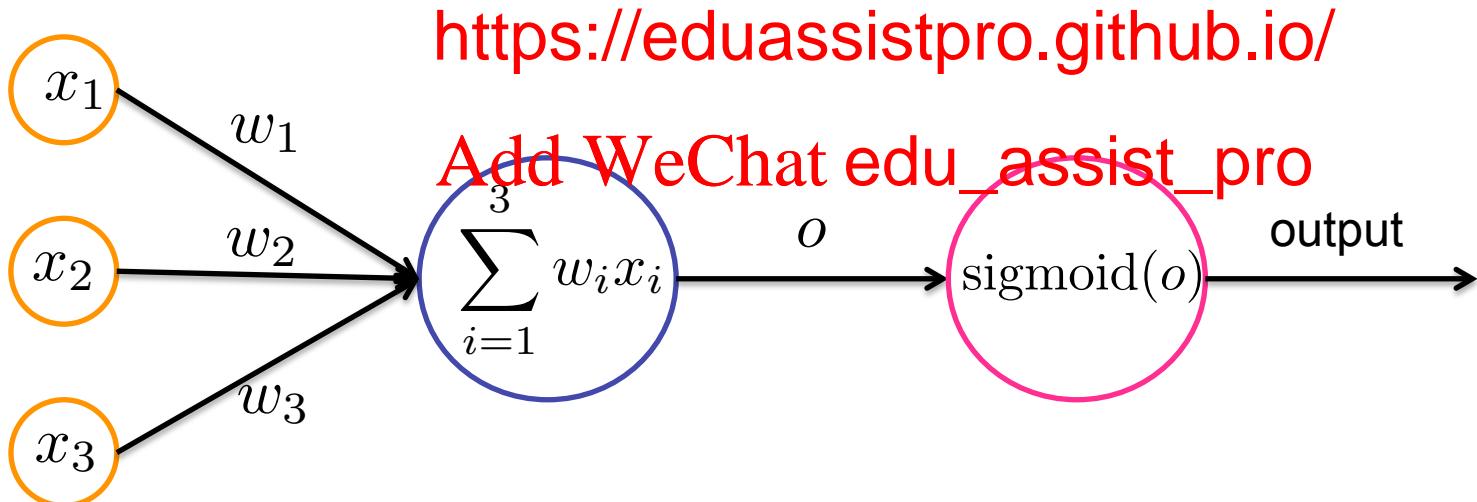


# Activation Function

- Sigmoid function

$$\text{sigmoid}(x) = \frac{1}{1 + \exp(-x)}$$

Assignment Project Exam Help



# Activation Function

---

- Hyperbolic tangent

$$\tanh(x) = \frac{\exp(x) - \exp(-x)}{\exp(x) + \exp(-x)}$$

Assignment Project Exam Help

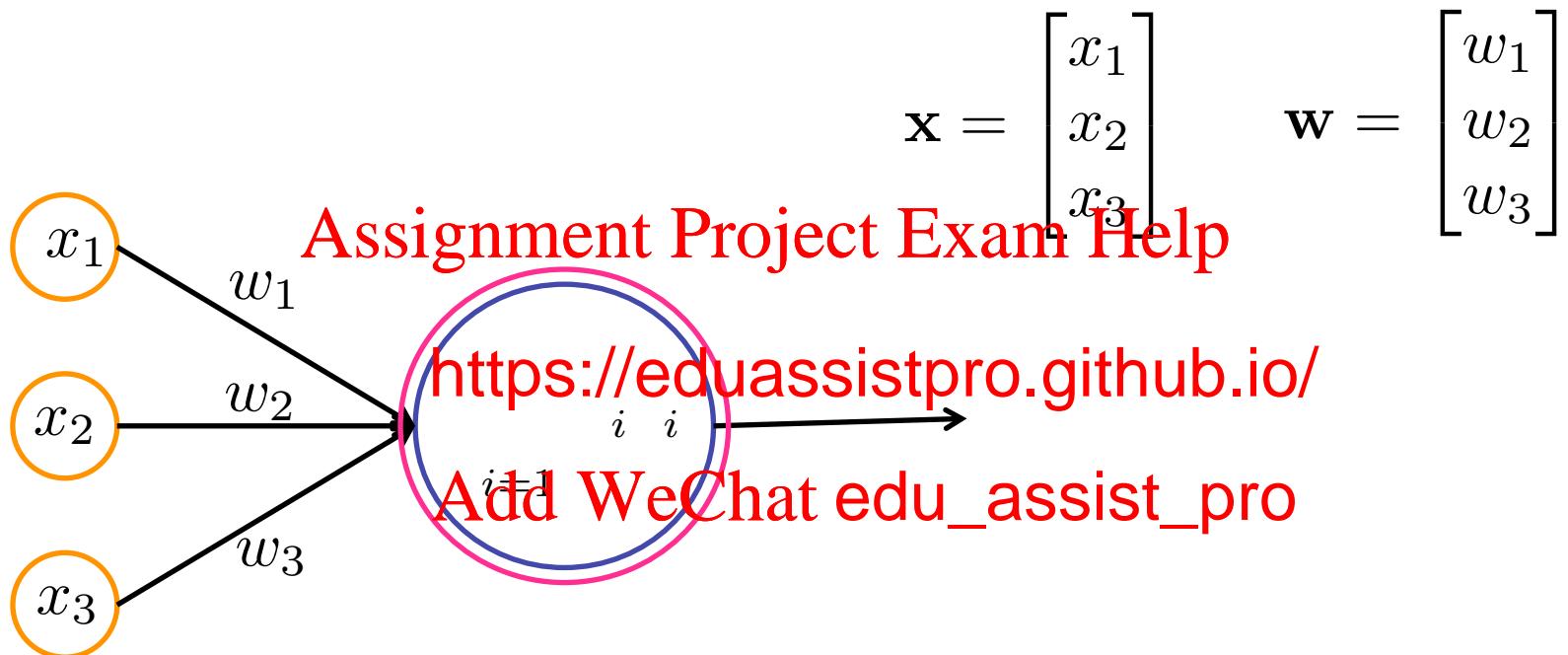
- Linear rectifie

<https://eduassistpro.github.io/>  
Add WeChat edu\_assist\_pro

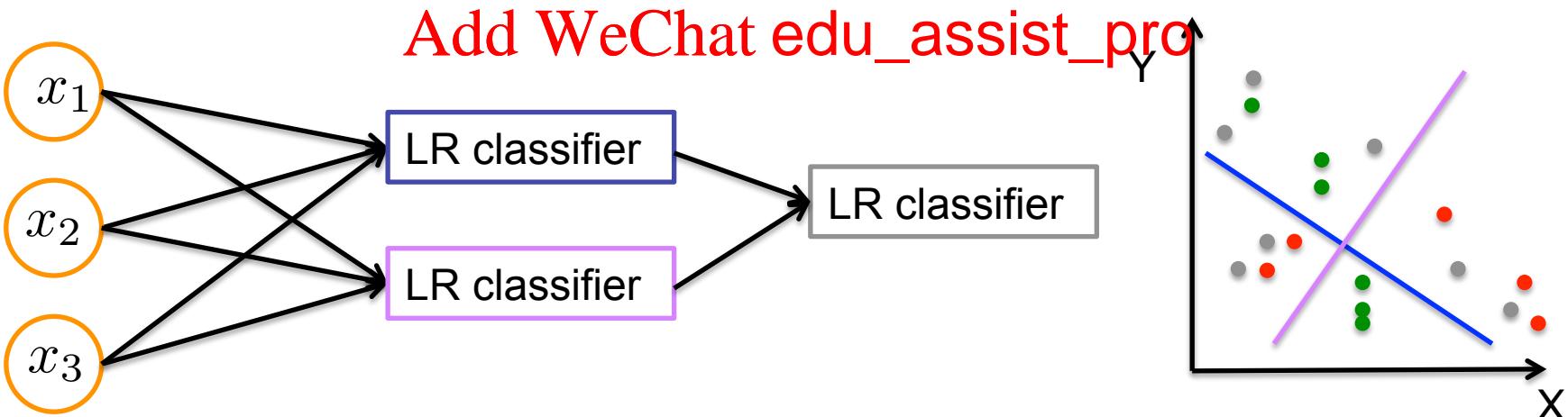
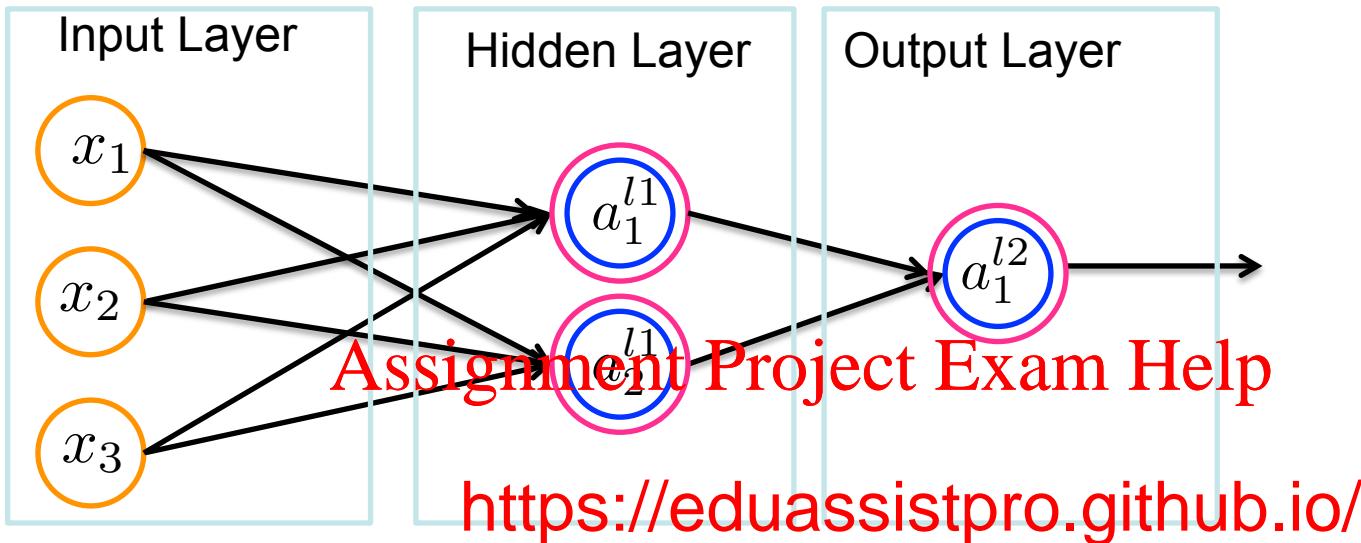
$$\text{rectifier}(x) = \max(0, x)$$

# Artificial Neuron (Compact form)

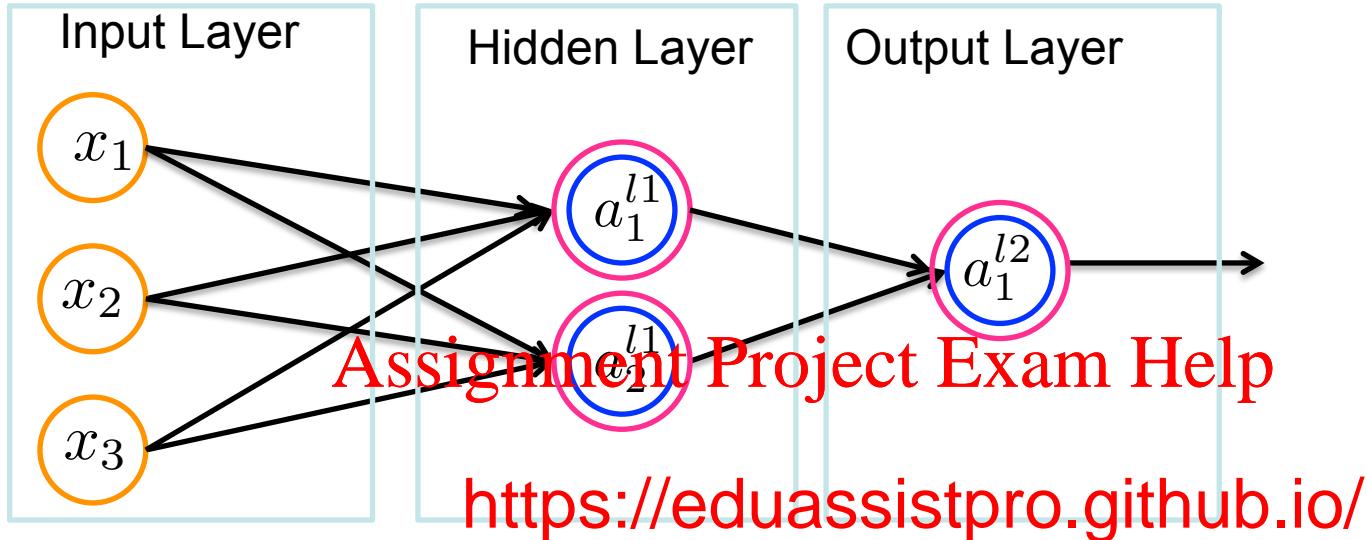
---



# Feedforward Networks



# Forward Propagation



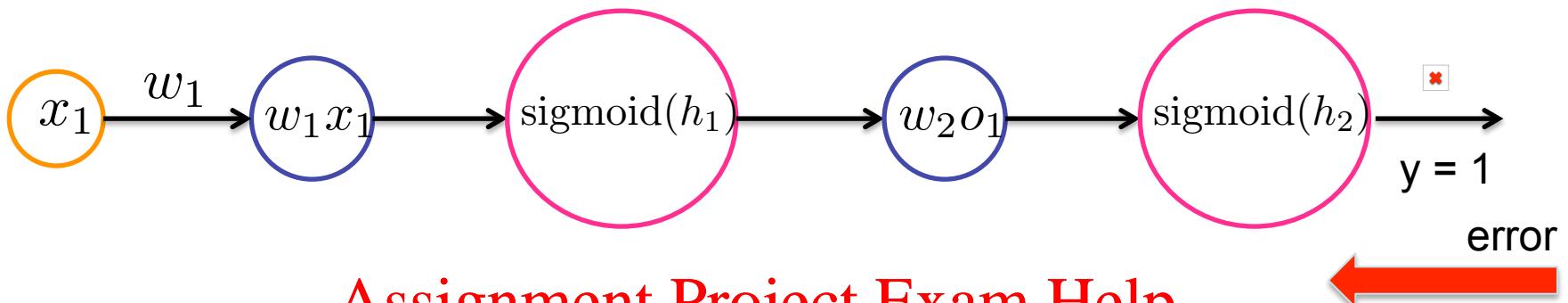
$$a_1^{l1} = g(w_{11}^{l0}x_1 + w_{21}^{l0}x_2 + w_{31}^{l0}x_3)$$
$$a_2^{l1} = g(w_{12}^{l0}x_1 + w_{22}^{l0}x_2 + w_{32}^{l0}x_3)$$

Add WeChat **edu\_assist\_pro** → 1       $g(w_{12}^{l1}a_1^{l1} + w_{22}^{l1}a_2^{l1})$

Composite Function:  $\mathbf{a}_1^{l2} = g(\mathbf{W}^{l1}g(\mathbf{W}^{l0}\mathbf{x}))$

Forward propagation

# Computing Derivatives



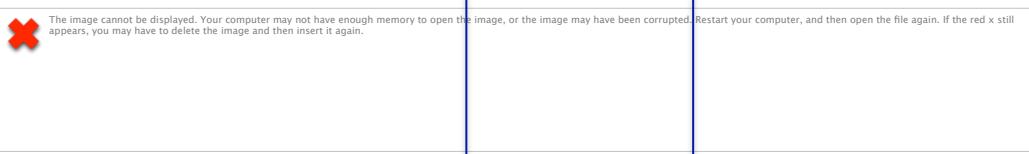
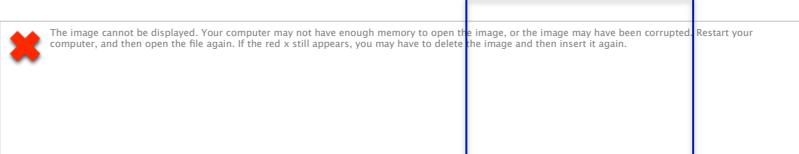
The image cannot be displayed. Your computer may not have enough memory to open the image, or the image may have been corrupted. Restart your computer, and then open the file again. If the red x still appears, you may have to delete the image and then insert it again.

The image cannot be displayed. Your computer may not have enough memory to open the image, or the image may have been corrupted. Restart your computer, and then open the file again.

<https://eduassistpro.github.io/>

The image cannot be displayed. Your computer may not have enough memory to open the image, or the image may have been corrupted. Restart your computer, and then open the file again. If the red x still appears, you may have to delete the image and then insert it again.

Add WeChat edu\_assist\_pro



# Overview of the NLP Lectures

---

- Introduction to natural language processing (NLP).
- Regular expressions, sentence splitting, tokenization, part-of-speech tagging.  
**Assignment Project Exam Help**
- Language mod  
**<https://eduassistpro.github.io/>**
- Vector semantics.  
**Add WeChat edu\_assist\_pro**
  - Multiclass logistic regression
  - Feedforward neural networks.
  - Word embeddings.
- Parsing.
- Compositional semantics and NLP applications.

# Weaknesses of Discrete Representations

---

- Missing new words.
- Hard to compute word similarity.
  - Similarity between Canberra and Paris?
- Require human labor to create  
<https://eduassistpro.github.io/>

# Distributional Similarity

---

“You shall know a word by the company it keeps”

(J. R. Firth 1957: 11)

## Assignment Project Exam Help

'study in the united st

e United States',  
<https://eduassistpro.github.io/>

'live in New Zealand',

australia.',

'study in the january 10'

Add WeChat edu\_assist\_pro  
on 05 January',

'stay in the New England',

ting on 01 December',

'live in the US',

'a meeting on January NUM',

'study in the United Kingdom'

'ordinary meeting on 9 December',  
'regular meeting of February 10'

# Word Co-occurrence Counts

---

Assignment Project Exam Help

<https://eduassistpro.github.io/>

Add WeChat edu\_assist\_pro

# Neural Word Representations

---

- Using continuous-value vectors.
- Learned from unlabeled data.
- Capture distributional similarity.

Assignment Project Exam Help

<https://eduassistpro.github.io/>

Add WeChat edu\_assist\_pro

# Learning Word Representation

---

Paris – France + Italy = Rome

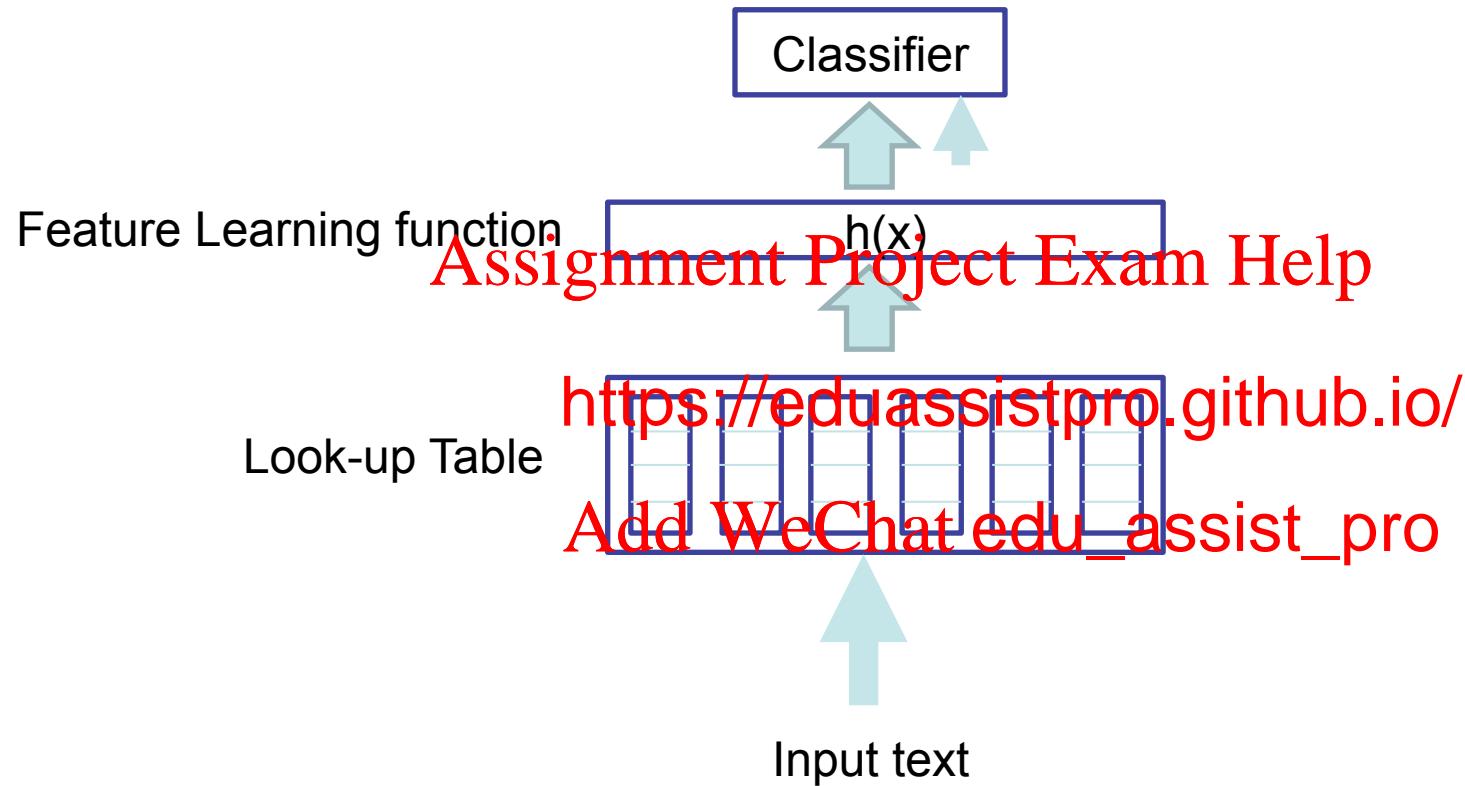
Android – Google + Microsoft = Windows

Assignment Project Exam Help

<https://eduassistpro.github.io/>

Add WeChat edu\_assist\_pro

# Neural Networks with Word Embeddings



# Learning Word Embeddings

---

- Predict the word in the middle given context words.

$$\prod_{i=1}^N P(x_i | x_{i-k}, \dots, x_{i-1}, x_{i+1}, \dots, x_{i+k})$$

<https://eduassistpro.github.io/>

*k* is the size of the con

Add WeChat edu\_assist\_pro

# Continuous Bag-of-Words Model (CBOW) [1,2]

- Basic form:  $P(x_i|x_{i-c}, \dots, x_{i-1}, x_{i+1}, \dots, x_{i+c}) = \frac{\exp(\mathbf{e}_i^T \mathbf{e}_{\text{sum}})}{\sum_{j \in V} \exp(\mathbf{e}_j^T \mathbf{e}_{\text{sum}})}$   
where  $c$  is the context size and  $V$  is the vocabulary.

Let  $c = 2$

softmax layer :

Assignment Project Exam Help

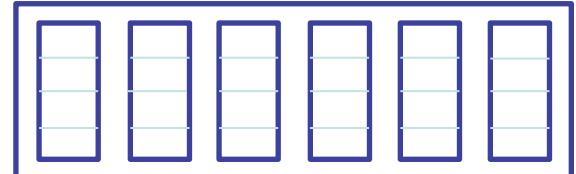
$$\frac{\exp(\mathbf{e}_i^T \mathbf{e}_{\text{sum}})}{\sum_{j \in V} \exp(\mathbf{e}_j^T \mathbf{e}_{\text{sum}})}$$

<https://eduassistpro.github.io/>

$$\mathbf{e}_{i-2} + \mathbf{e}_{i-1} + \mathbf{e}_{i+1} + \mathbf{e}_{i+2}$$

Add WeChat edu\_assist\_pro  
 $(\mathbf{e}_{i-2}, \mathbf{e}_{i-1}, \mathbf{e}_i, \mathbf{e}_{i+1}, \mathbf{e}_{i+2})$

Look-up table:

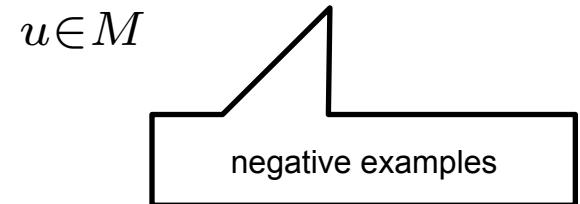
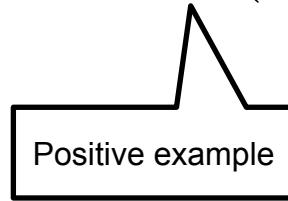


$$(x_{i-2}, x_{i-1}, x_i, x_{i+1}, x_{i+2})$$

# High Scalability with Negative Sampling

- Approximate the softmax loss with a set of binary classification losses.
  - Positive example: the loss of seeing word  $i$  given context words.
  - Negative examples : the loss of not observing  $|M|$  word  $j$  given context words from  $\cup_{u \in M}$  of randomly picked words from  $v$

$$L_{\text{neg\_sam}}(x_i) = -\log \sigma(\mathbf{e}_i^T \mathbf{e}_s) + \log(1 - \sigma(\mathbf{e}_u^T \mathbf{e}_{\text{sum}}))$$



$$\text{where } \sigma(z) = \frac{1}{1+\exp(-z)}$$

# Skip-Gram [1,2]

- Basic form:  $\prod_{-c \leq j \leq c, j \neq 0} P(x_{i+j}|x_i) = \prod_{-c \leq j \leq c, j \neq 0} \frac{\exp(\mathbf{e}_{i+j}^T \mathbf{e}_i)}{\sum_{l \in V} \exp(\mathbf{e}_l^T \mathbf{e}_i)}$

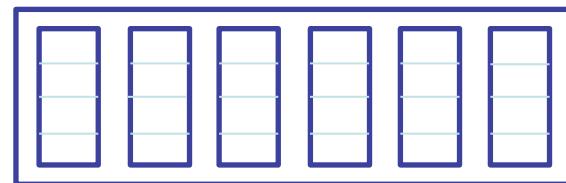
where  $c$  is the context size and  $V$  is the vocabulary.

Assignment Project Exam Help  
softmax layer:  $\frac{\exp(\mathbf{e}_{i+j}^T \mathbf{e}_i)}{\sum_{l \in V} \exp(\mathbf{e}_l^T \mathbf{e}_i)}$

<https://eduassistpro.github.io/>

Add WeChat edu\_assist\_pro

Look-up table:



$(x_{i-2}, x_{i-1}, x_i, x_{i+1}, x_{i+2})$



# Skip-gram with Negative Sampling

---

$$L_{\text{neg\_sam}}(x_i) = \sum_{\substack{-c \leq j \leq c, j \neq 0}} [ -\log \sigma(\mathbf{e}_{i+j}^T \mathbf{e}_i) - \sum_{u \in M_j} \log(1 - \sigma(\mathbf{e}_u^T \mathbf{e}_i)) ]$$

Assignment Project Exam Help

<https://eduassistpro.github.io/>

negative examples

Add WeChat edu\_assist\_pro

$$\text{where } \sigma(z) = \frac{1}{1 + \exp(-z)}$$

# Evaluation with Word Analogy [1]

---

$$\mathbf{e}_? = \mathbf{e}_{\text{Athens}} - \mathbf{e}_{\text{Greece}} + \mathbf{e}_{\text{Norway}}$$

Assignment Project Exam Help

<https://eduassistpro.github.io/>

Add WeChat edu\_assist\_pro

# TensorFlow

---

- Tensors.
  - scalar (rank 0) : 1.8, 2 etc.
  -

Assignment Project Exam Help

<https://eduassistpro.github.io/>

Add WeChat edu\_assist\_pro

- Define functions on tensors.
- Auto-differentiation.

# Computation Graph

---

- Nodes
  - tensors.
  - tensor operations.

Assignment Project Exam Help

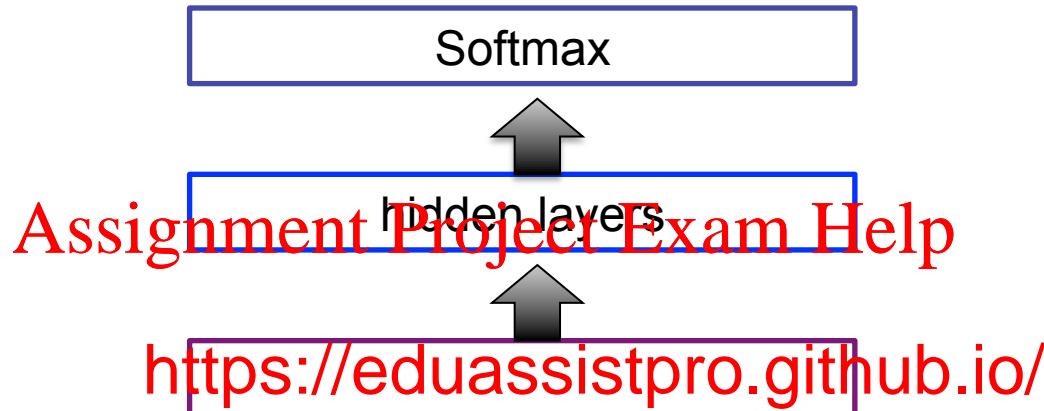
<https://eduassistpro.github.io/>

Add WeChat edu\_assist\_pro

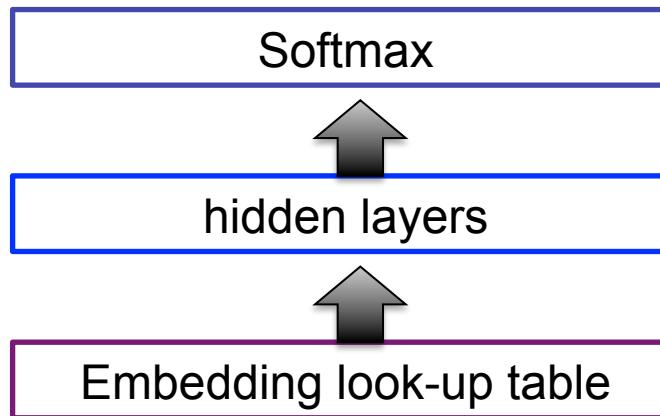
# Applications

---

- Neural Language Model [4].



- Sentiment analysis [5, 6].



# REFERENCES

---

- [1] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient Estimation of Word Representations in Vector Space. In Proceedings of Workshop at ICLR, 2013.
- [2] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. Distributed Representations of Words and Phrases and their Compositionality. In Proceedings of NIPS, 2013.
- [3] <https://code.go>
- [4] Yoshua Bengi <https://eduassistpro.github.io/> c/  
Vincent and Christian Jauvin. A Neural Probabilistic Lan  
JMLR, 2003.
- [5] Yoon Kim. Convolutional Neural Sentence  
Classification.
- [6] Richard Socher, Alex Perelygin, Jean Y. Wu, Jason Chuang,  
Christopher D. Manning, Andrew Y. Ng and Christopher Potts.  
Recursive Deep Models for Semantic Compositionality Over a  
Sentiment Treebank.
- [7] Understanding Word Vectors.  
<https://medium.com/explorations-in-language-and-learning/understanding-word-vectors-f5f9e9fdef98>